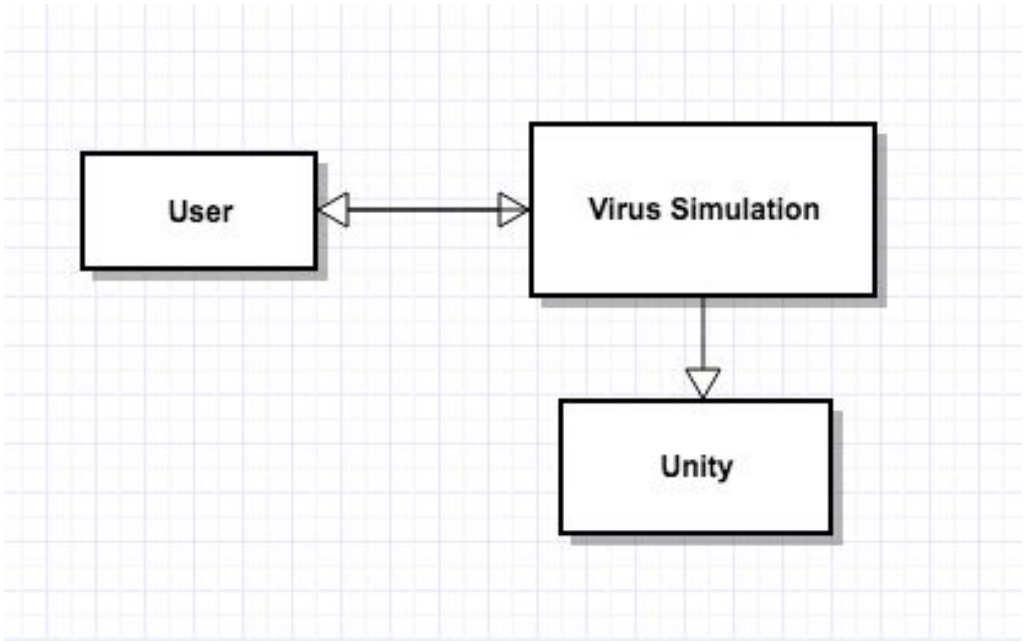
Architectural Design

Epidemiology Group

Conceptual diagram:



Architectural design decisions:

There is a generic application architecture that can act as a template for the system, which is for applications.

The system will most likely use multiple cores to process the complex simulations that must be done in real time.

For architectural patterns we intend to use MCV and layered.

The fundamental approach used to structure the system will be simplicity.

The structural components will be divided into sub-components by their relation to each other, with parent systems being above the child systems that depend on them.

The best architectural organization for our non-functional requirements should be layered information system architecture.

The system architecture of the system will be documented with doxygen which will help cut down costs on development.

The strategy to control the operation of components on the system will be to use the unity engine.

We used the website Gliffy for our diagrams.

Architecture and System characteristics:

(1). Performance: performance affected the structure of the system by influencing how different modules interact with one another, as well as by enforcing that code runs efficiently so that the simulation can run on a large range of systems.

(2). Security: There is no part of our program that would require security as there is no sensitive information that could be gained. Our game does not contain critical assets that are dependent on a third party.

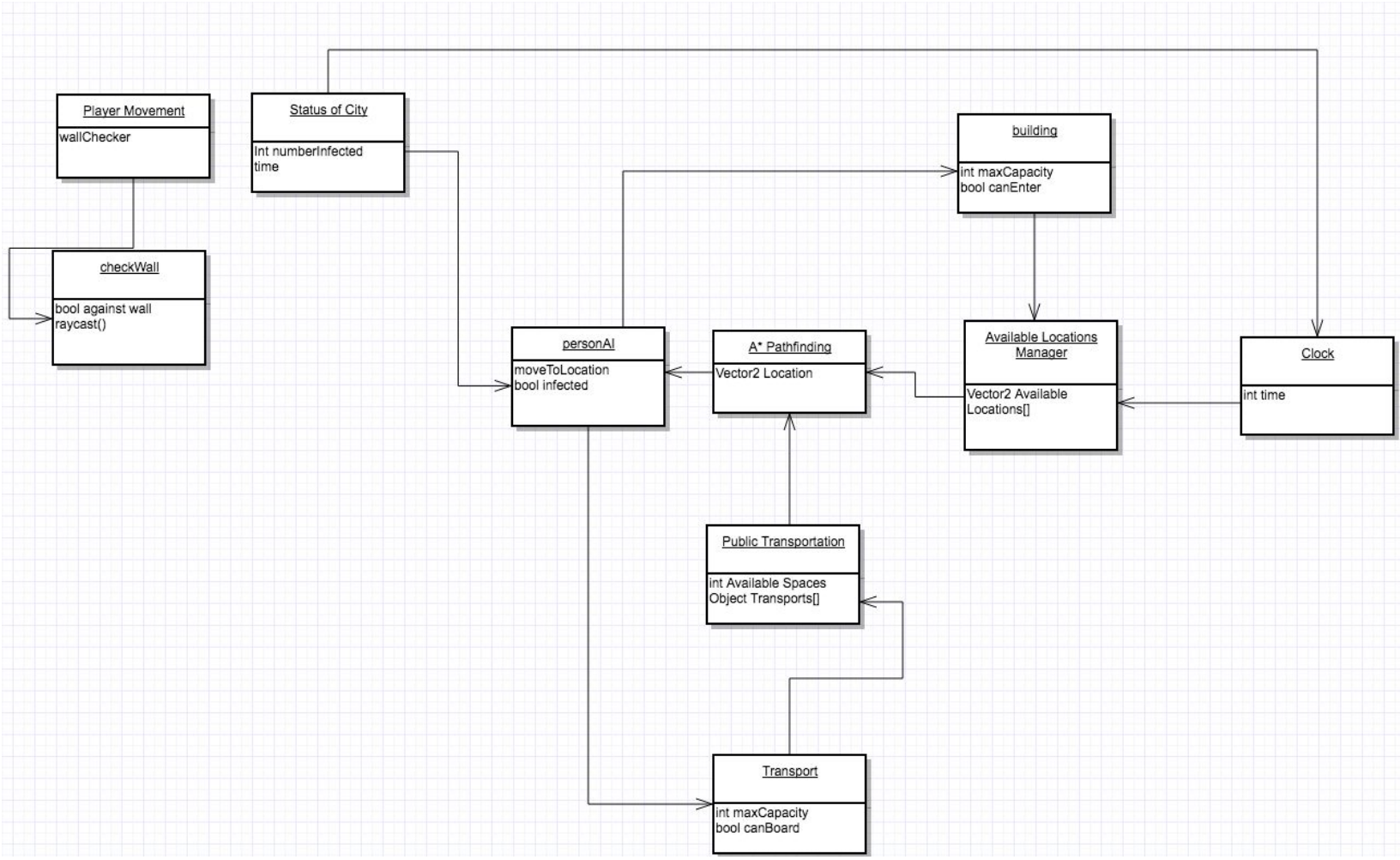
(3). Safety: The program is stand alone and would not affect other parts of the system.

(4). Availability: It should function on Windows, Mac, and Linux Personal computer systems.

(5). Maintainability: This characteristic had a strong influence on our system architecture because it forced components to be independent of one another so that they can be replaced easily.

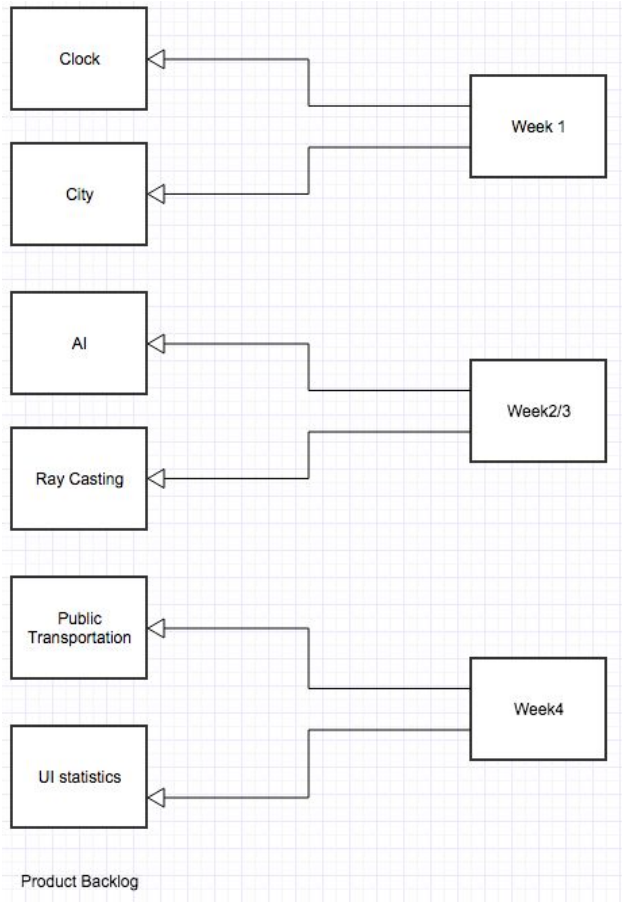
4+1 view:

Logical View:

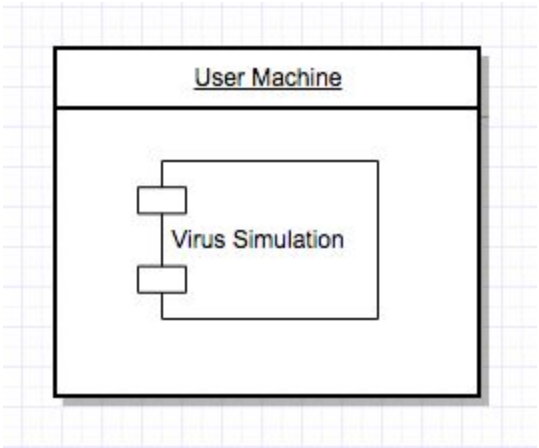


Process View:

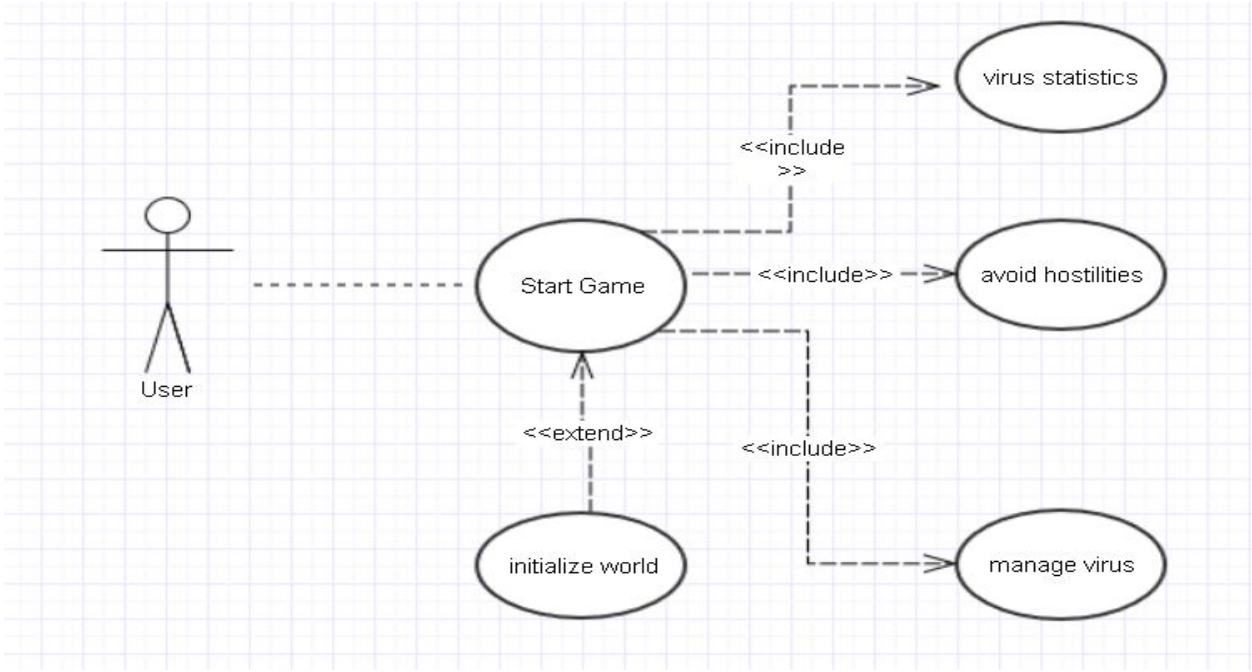
Development View:



Physical View:



Scenario View:



Architectural Patterns: You can see MVC(Model-View-Controller) since the display and keyboard are separate from one another. The keyboard is our controller and will take in the user inputs to move around and interact with objects within the simulation. The View is the monitor, the user will watch the simulation through it. We did use this architecture for this project since most games use this architecture and the hope is to make this simulation into an interactive game if the user wishes.